

Domain-Producer Detailed Specification

1. Purpose and Role

The purpose of the Domain-Producer is to generate concrete outputs (code, configuration files, or packaged artifacts) from the project specification and domain rules. It is the main workhorse of the architecture; it is the one that generates the output that will need validation and post processing.

It is a Domain Related Agent, working after the Domain-Speculus has defined the rules. It consumes the ARCOSProject element, which contains domain vocabulary, domain schema, and rules, and generates a packaged output that can then be validated and post-processed if needs be.

Unlike the ARCOS-Speculus or Domain-Speculus, the Producer does not interact directly with the user. Instead, it communicates only with the Orchestrator via schema-based XML messages.

The producer can either produce the desired output or ask for clarifications. Upon clarification, the Orchestrator sends the question back to the Domain-Speculus to ask the user to respond to the clarification by adding some rules so the producer can continue. The Domain-Speculus send back the new rules to the Orchestrator that sends it back with a retrial. If the Producer never produced any output yet, the Orchestrator treats it as a new call to the Producer, without sending back the previous output since it never got produced.

This is where ARCOS shines ... it does not care about the domain ... it can produce whatever you want ... if it needs AI to do it ... and it needs validation and post processing ! Be it some legal documents that needs to have certain clauses in it ... If you have Domain Related Agents for it, it will be possible to have a process where you produce, validate and post process it.

2. Process Flow

Once the Maestro receives the response from the Domain Speculus, it sends the ARCOSProject element of the response to the Domain Producer, which contains project's purpose, the domain vocabulary and rules, and the domain schema.

In case of retries, the Orchestrator also sends the previous output that the Domain Producer previously generated, as well as the report that shows it failed.

The Domain Producer can ask for clarification before executing its process, sending the clarification request back to the Orchestrator so that it can send the clarification to the Domain Speculus to try to find new rules to clarify the issue. The Domain Speculus then sends back the new rules to the Orchestrator that sends back a request to the Domain Producer with the new rules in the ARCOSProject element it received from the Domain Speculus.

The Domain Producer generates the required output, be it code, contracts, or other deliverables, then packages it into an OutputZipBase64 element to be sent back to the Orchestrator. It is the job of the Artefact Extractor to hold copies of each part of the output so the Orchestrator can, on retries, send only the related producer parts found in the error reports back to the producer to correct them (the job of the Focus Extractor).

The Domain-Producer serves to generate the concrete outputs of the project specification, acting as the main generator within the ARCOS orchestration loop.

3. Communication between the Domain Producer and the Orchestrator (Maestro) using schema based XML

Maestro_Producer_Messaging.xsd

<ProducerRequest>		
<InitialProduction> or	Choice	
<ARCOSProject>	Required	common:InlineArtifactType
<RetryProductionType>	Choice	
<ARCOSProject>	Required	
<OutputZipBase64>	Required	common:InlineArtifactType
<ValidationReport> or	Choice	common:InlineArtifactType
<PostProcessorReport>	Choice	common:InlineArtifactType
<Checksum>	Required	common:SHA256Type

The integrity of the package is verified by SHA-256 checksum

Producer_Response.xsd

<ProducerResponse>		
<OutputZipBase64> or	Choice	common:InlineArtifactType
<Clarification>	Choice	
<ProjectID>	Required	
<Questions>	Required	
<Question>	Required	xs:string

4. Dependencies and Inputs

The Domain-Producer uses the following:

- Project's Purpose
- Domain_Vocabulary.xml (entities, attributes, enums, units, phrases)
- Predefined_Domain_Rules.xml (global rules)
- Current_Project_Rules.xml (user-extracted and accepted rules)
- Domain Schema (XSD for the domain's data model)

These are provided inline in ARCOSProject element of the request.

5. Outputs and Artifacts

The Producer outputs are packaged as a Zip file encoded in base64. This is returned in the Output element of ProducerResponse. The integrity of the package is verified by SHA-256 checksum. Artifacts can be further described using ArtifactManifest.xml, which tracks files, their paths, and history metadata (producer, version, timestamps).

6. Failure and Retry Handling

If validation fails, the Orchestrator issues a RetryProduction request with additional context (Validator or Post-Processor reports, plus previous output). The Producer must handle retries deterministically, generating revised outputs based on the provided feedback.

7. Constraints and Licensing

The Domain-Producer must respect the project's constraints as defined in `ARCOS_Project.xml`:

- Performance: level
- Security: encryption, policy
- Realtime / Embedded / Portable flags

Licensing information (mode, validation, duration) is also passed in `ARCOS_Domain_Agents.xml` and governs how the Producer may be used.